

## Windows alapok

Operációs rendszerek gyakorlati útmutató

Készítette: Micskei Zoltán

Utolsó módosítás: 2014.02.19.

A gyakorlat célja, hogy előkészítsünk egy Windowst futtató virtuális gépet, amit majd a félév folyamán használni tudunk, valamint azon egy pár alapvető eszközt kipróbáljunk.

A tárgyban Windows 8.1-et fogunk használni, mivel már elérhető a végleges változat belőle.

### 1 Virtuális gép elkészítése

A gyakorlatok javasolt virtuális gépeken végrehajtani, hogy ne az éles fizikai környezetünket állítsuk el.

#### 1.1 Virtualizációs keretrendszerek

Első lépésként egy virtualizációs keretrendszert érdemes választani, itt is több lehetőségünk van:

- *VMware Player*<sup>1</sup>: a VMware ingyenes terméke, Windowsra és Linuxra is elérhető. A 3-as verziótól kezdve ez is tud virtuális gépeket létrehozni. A 4-es verziótól kezdve 64 bites CPU szükséges. A tanszéken ezt használjuk, ehhez elérhető egy részletes leírás, amit érdemes lehet átolvasni [1].
- *VirtualBox*<sup>2</sup>: nyílt forráskódú megoldás, amit az Oracle támogat. Windowsra, Linuxra és Mac OS X operációs rendszerekre is elérhető.
- *stb.*

Mindegyik megoldásban közös, hogy 64 bites vendéget csak akkor tudnak futtatni, ha a fizikai gép processzorában van hardveres virtualizációs támogatás (Intel VT-x vagy AMD-V). Ha nem vagyunk benne biztosak, hogy rendelkezik-e a CPU-nk ilyen kiegészítéssel, akkor ezt például a CPU-Z<sup>3</sup> segédprogrammal vagy Linuxon a `/proc/cpuinfo` tartalmában a `flags` résznél tudjuk ellenőrizni.

#### 1.2 Windows telepítőkészlet beszerzése

Több forrásból is elérhető Windows kipróbálás és tanulás céljából:

---

<sup>1</sup> VMware Player, <http://www.vmware.com/products/player>

<sup>2</sup> VirtualBox, <https://www.virtualbox.org/>

<sup>3</sup> CPU-Z, <http://www.cpuid.com/softwares/cpu-z.html>

Próbaverzió: A Microsoft oldaláról is elérhető egy 90 napos próbaverzió<sup>4</sup>. Ez ISO formában tölthető le, a próbaverzió pontos feltételeit érdemes elolvasni az oldalon.

**DreamSpark (MSDNAA)** Már a Windows 8.1 is elérhető a korábbi DreamSpark (korábbi nevén MSDNAA<sup>5</sup>) előfizetésének keretében. Az MSDNAA programba regisztrálni kell, majd ezután az OnTheHub által üzemeltetett letöltő szerverről lehet megrendelni az egyes termékeket (a regisztráció néhány napos átfutással rendelkezik). Az MSDNAA keretében személyre szabott kulcsokat kapunk, amiket csak véges alkalommal lehet aktiválni. Ezért érdemes először nem aktiválni a csak teszteléshez használt virtuális gépet, az alpból 30 napig működik aktiválás nélkül, ami meghosszabbítható párszor (lásd az `s1mgr` parancs `/dlv` és `/rearm` kapcsolóit [2]).

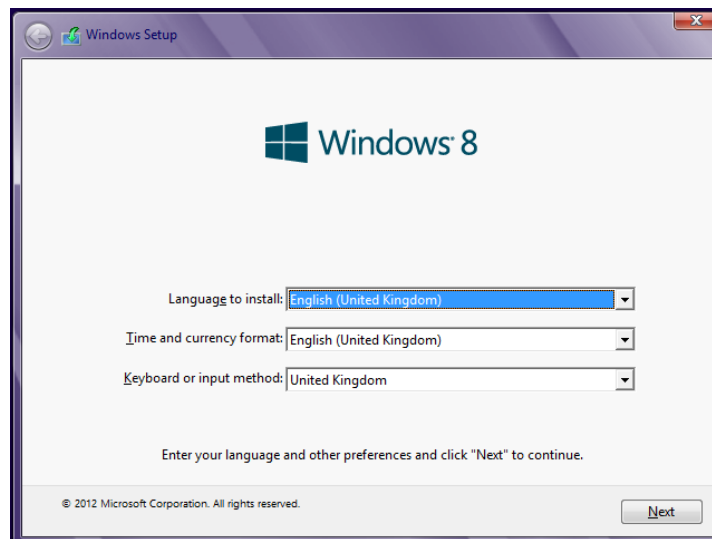
### 1.3 A Windows telepítése

Maga a telepítés folyamata viszonylag magától értetődő. A virtuális gépnek érdemes legalább 1 GB RAM-ot beállítani, ezen kívül kezdetben kb. 10 GB helyet igényel, később ez 15 GB-ra megnőhet, ahogy használjuk a virtuális gépet.

Javasolt, hogy angol nyelvű változatot telepítsünk.

A következőkben a Windows 8 telepítési lépéseit nézzük gyorsan végig (a Windows 8.1 is nagyon hasonló). A virtuális gép virtuális CD-meghajtó beállításainál adjuk meg a telepítő ISO-t, ezután a virtuális gép indulásakor erről fog bootolni.

A telepítő a következő képernyővel üdvözl (1. ábra).



1. ábra: Windows 8 telepítő

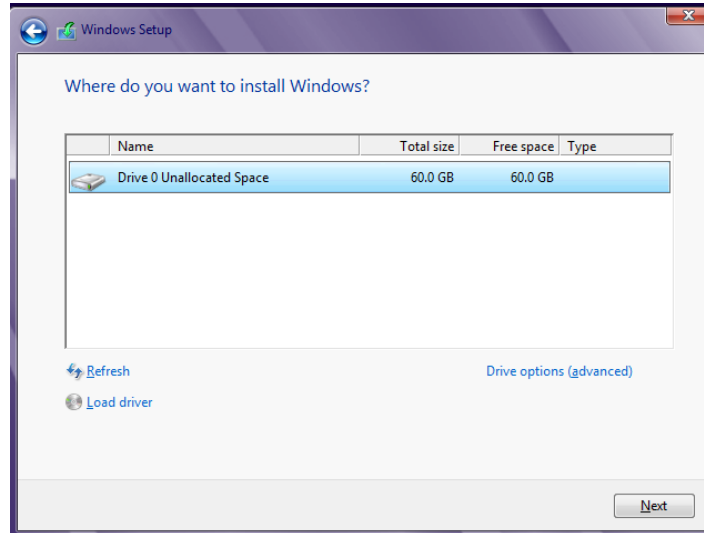
A billentyűzetkiosztás beállítása egyéni ízlés kérdése.

A licenfeltételek elfogadása után válasszuk ki, hogy hova telepítjük az operációs rendszert (2. ábra). Mivel itt nincs korábbi rendszer, és csak egy merevlemez tartozik a virtuális

<sup>4</sup> Windows 8.1 evaluation, <http://technet.microsoft.com/en-us/windows/hh771457.aspx>

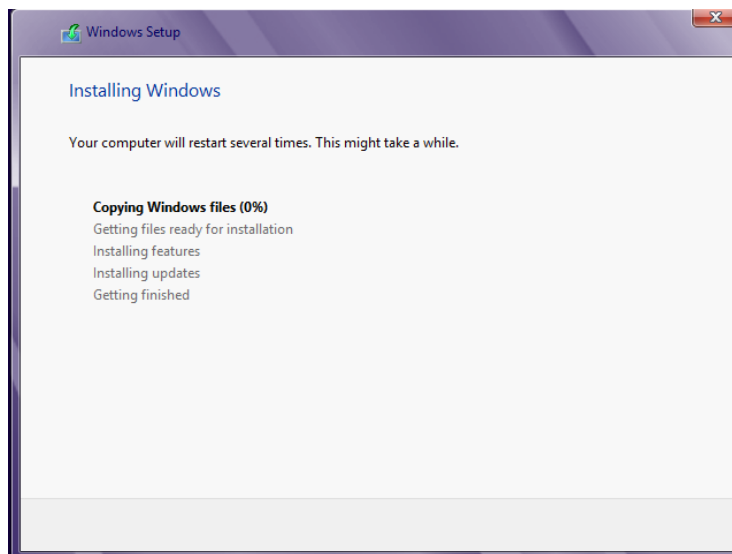
<sup>5</sup> BME MSNDAA, <http://msdnaa.bme.hu/>

géphez, ezért semmi különösebb teendők nincsen. A *Next* megnyomása után a telepítő a lemezen létrehoz két partíciót. Egy kisebb méretű boot és helyreállítási célokra, a lemez további részét pedig egy nagy partícióval tölti ki (az operációs rendszernek és az adatainknak). A *Drive options* gomb megnyomásával saját magunk végezhetnénk el a partíciók létrehozását.



**2. ábra: Telepítés helyének kiválasztása**

Ezután nincs más teendőnk, mint várni, hogy a telepítés véget érjen.



**3. ábra: A tényleges telepítés**

A telepítés után még a kezdeti beállításokat kell elvégezni (pl. felhasználónév megadása).

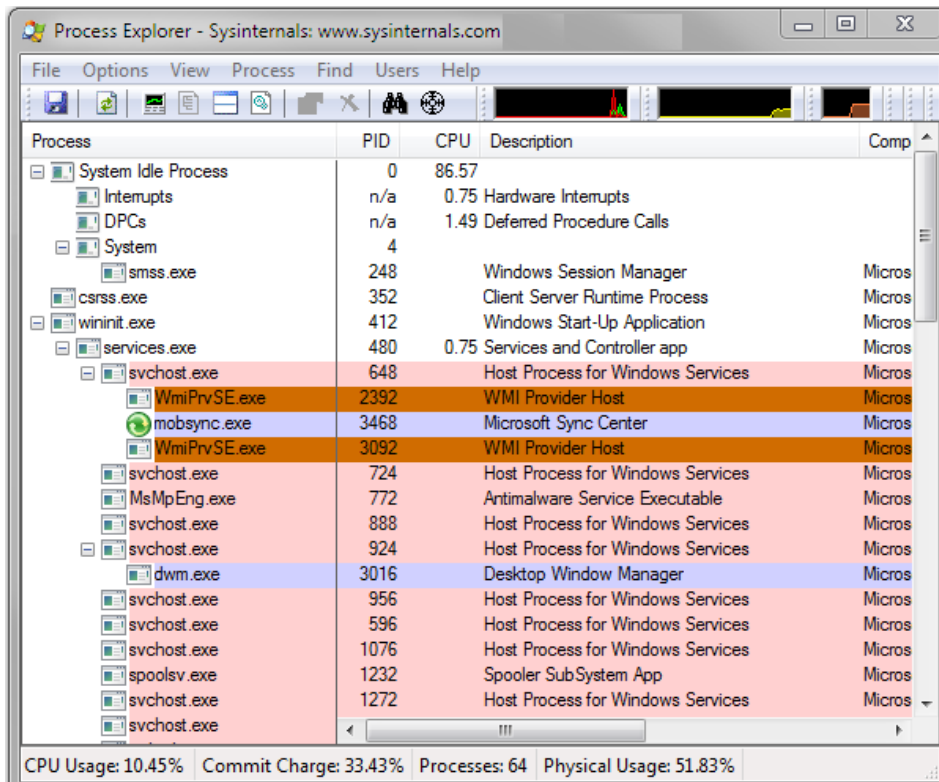
## 2 Hasznos segédprogramok beszerzése

A beépített segédeszközökön kívül, pl. Feladatkezelő (Task Manager) vagy Erőforrás-figyelő (Resource Monitor), érdemes még néhány segédeszközt kezdetben feltelepíteni a virtuális gépre.

### 2.1 Sysinternals Suite

A Sysinternals cég hasznos kis segédprogramokat készített, amikkel a Windows belső működését lehetett tanulmányozni, vagy a hibakeresésben segítettek. A Microsoft megvásárolta a céget, de az eszközök továbbra is elérhetőek ingyenesen<sup>6</sup>, azokat folyamatosan fejlesztik. Érdemes az eszközök nagy részét tartalmazó *Sysinternals Suite* csomagot letölteni.

Először a *Process Explorer* eszközzel ismerkedjünk meg, ez a Feladatkezelő „okosabb” változata (4. ábra).



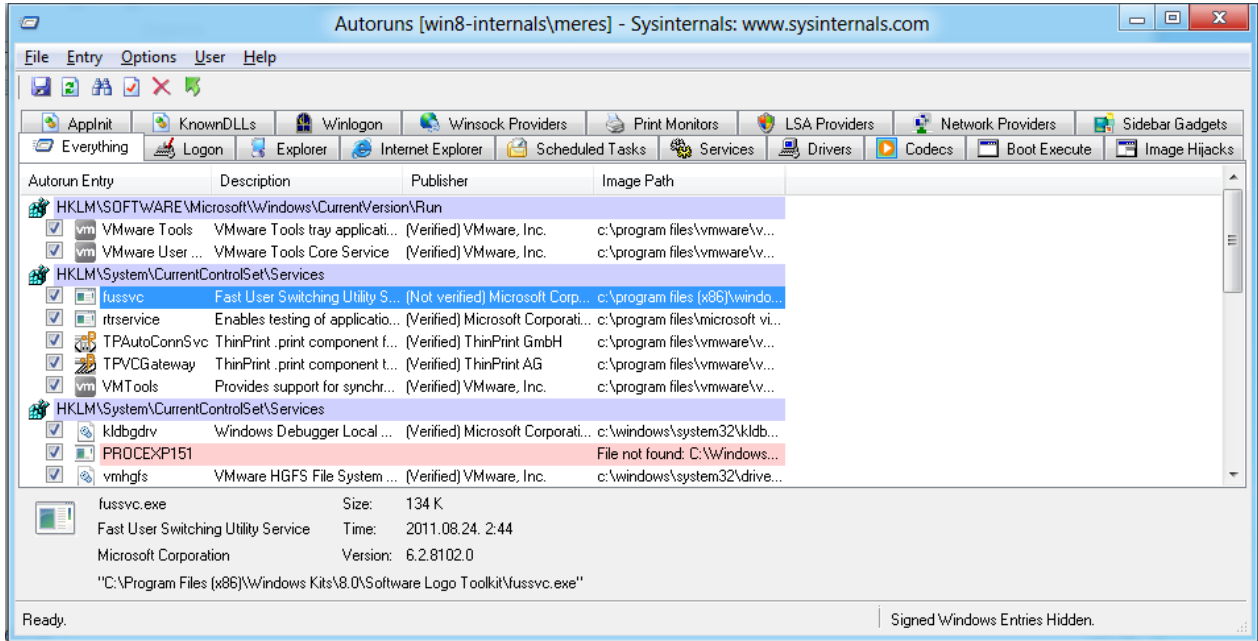
4. ábra: A Process Explorer felülete

**Feladat:** nézzük meg, hogy milyen folyamatok futnak a frissen feltelepített Windows virtuális gépünkben, és nézzünk utána, ha valamelyik neve nem ismerős.

Egy másik gyakran használt eszköz az *Autoruns*, ami megmutatja, hogy milyen programok és kiegészítők indulnak el automatikusan a rendszer indulásakor (5. ábra). Az *Options* menüben azt is be lehet állítani, hogy ellenőrizze a programok digitális aláírását.

<sup>6</sup> Sysinternals, <http://technet.microsoft.com/en-us/sysinternals>

**Feladat:** nézzük meg, hogy milyen programok és szolgáltatások indulnak el automatikusan a virtuális Windowsunk indulásakor.

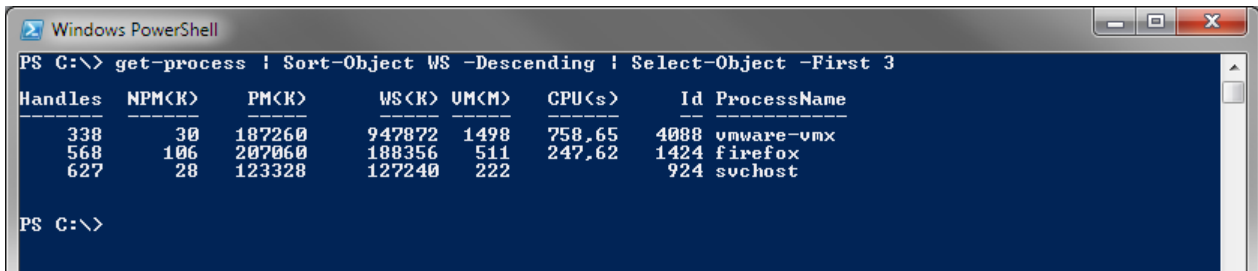


5. ábra: Lista az automatikusan induló programokról az Autorunsban

Érdeemes ezután Sysinternals weboldalán végigfutni az eszközök listáján, nagyon hasznos segédprogramok érhetőek el (pl. a TcpView a nyitott hálózati kapcsolatokat jeleníti meg, a BglInfo a rendszer legfontosabb adatait jeleníti meg háttérképként stb.).

## 2.2 PowerShell

A PowerShell a Microsoft új parancssori shell környezete. Egyesíti a szkriptnyelvek egyszerűségét és a .NET objektum-orientált szemléletét és kiterjedt osztálykönyvtárát. Érdeemes megismernedni vele, mert ismétlődő feladatok (pl. sok fájl átnevezése, adott tartalom periodikus letöltése) automatizálására egy gyors eszköz.



6. ábra: PowerShell konzol

A PowerShell úgynevezett cmdleteket (Ige-Főnév formájú elnevezésekkel) biztosít a legváltozatosabb funkciók elérésére. Hajtsuk végre például a következő parancsot a PowerShell konzolban:

---

Get-Process | Sort-Object WS -Descending | Select-Object -First 3

Mit is csináltunk? Lekértük először a folyamatok listáját (Get-Process), sorba rendeztük őket a WS nevű attribútumok szerint (WS – working set, az aktuális fizikai memóriahasználat), majd végül az első hármat tartottuk csak meg (Select-Object). Más korábbi shell környezetekkel ellentétben itt az az igazán kényelmes, hogy nem strukturálatlan szöveg adódik át az egyes parancsok között, hanem típusos objektumok, így a szűrés és rendezés viszonylag könnyű.

**Feladat:** próbáljuk meg lekérdezni az első öt futó folyamatot, ami a legtöbb CPU időt használja!

Bár a tantárgyban nem kerül majd elő részletesebben, de aki Windowst használ, annak érdemes legalább alapszinten megismerkedni a PowerShell környezettel [3].

### 2.3 Fejlesztői környezet

Ha szeretnénk megismerkedni a Windows alacsonyabb szintű programozási felületeivel, eszközmeghajtók készítésével, akkor érdemes felrakni a következőket is.

- **Visual Studio 2013 Express for Windows Desktop:** ingyenes fejlesztői környezet, mely tartalmazza a parancssori fordítókat és egy IDE-t is.
- **Windows SDK:** segédeszközök és header fájlok. A Windows 8-as verzió óta a fejlesztőkörnyezet nem része. A legfrissebb kiadott verziója a 8.1<sup>7</sup>.
- **Windows Driver Kit (WDK):** az eszközmeghajtók fejlesztéséhez szükséges könyvtárak, fordítók és keretrendszerek. A 8-as változattól kezdve integrálódik a Visual Studio IDE-vel, viszont legalább Professional változat kell a VS-ből ehhez. A 8.1-es változata ennek is letölthető<sup>8</sup>.
- **Windows Debugging Tools:** debugger eszközök natív kódhoz. Lehet kernel debugging módban is használni, amivel az operációs rendszer belső adatstruktúráihoz is hozzáférünk. A Windows SDK részeként érhető el.

## 3 Hivatkozások

- [1] Micskei Zoltán. „Virtuális gépek – VMware Player leírás”, Mérés labor 4. segédlet, 2013.  
URL: <http://www.mit.bme.hu/oktatas/targyak/vimia315/mindegyik-mereshez-wireshark-vmware-player>
- [2] Microsoft Technet. „Simgvbs Options”, URL: <http://technet.microsoft.com/en-us/library/ff793433.aspx>
- [3] PowerShell Pro. „PowerShell Tutorials”, URL: <http://www.powershellpro.com/powershell-tutorial-introduction/>

---

<sup>7</sup> Windows SDK, <http://msdn.microsoft.com/en-us/windows/desktop/bg162891.aspx>

<sup>8</sup> Windows Driver Kit, <http://msdn.microsoft.com/en-us/windows/hardware/gg454513.aspx>